

```

1. Serializable]
2.     [SqlUserDefinedType(Format.Structured, MaxByteSize=8000)]
3.     public class BaseItem: INullable
4.     {
5.         [SqlUdtField(IsNullable=false)]
6.         private SqlGuid m_ID;
7.
8.         [SqlUdtField(MaxSize=128, IsFixedLength=false)]
9.         private SqlString m_Name;
10.
11.        [SqlUdtProperty(FieldName="m_ID")]
12.        public SqlGuid ID
13.        {
14.            get
15.            {
16.                return m_ID;
17.            }
18.            set
19.            {
20.                this.m_ID = value;
21.            }
22.        }
23.
24.        [SqlUdtProperty(FieldName="m_Name")]
25.        public SqlGuid Name
26.        {
27.            get
28.            {
29.                return m_Name;
30.            }
31.            set
32.            {
33.                this.m_Name = value;
34.            }
35.        }
36.
37.        [SqlUdtField(IsNullable=true)]
38.        public MultiSet<PropertyAssociation> Properties;
39.
40.        #region UDT boilerplate
41.        public BaseItem()
42.        {
43.            this.ID = new SqlGuid(Guid.NewGuid());
44.        }
45.        public override string ToString()
46.        {
47.            return "ID " + this.ID;
48.        }
49.        [SqlUdtField]
50.        protected SqlBoolean m_IsNull = SqlBoolean.False;
51.        public bool IsNull { get { return this.m_IsNull.Value; } }
52.        public static BaseItem Null
53.        {
54.            get
55.            {
56.                BaseItem s = new BaseItem();
57.
58.                s.m_IsNull = SqlBoolean.True;
59.                return s;
60.            }
61.        }
62.        public static BaseItem Parse(SqlString s)
63.        {
64.            return new BaseItem();
65.        }
66.        #endregion
67.    }

```

Fig. 1

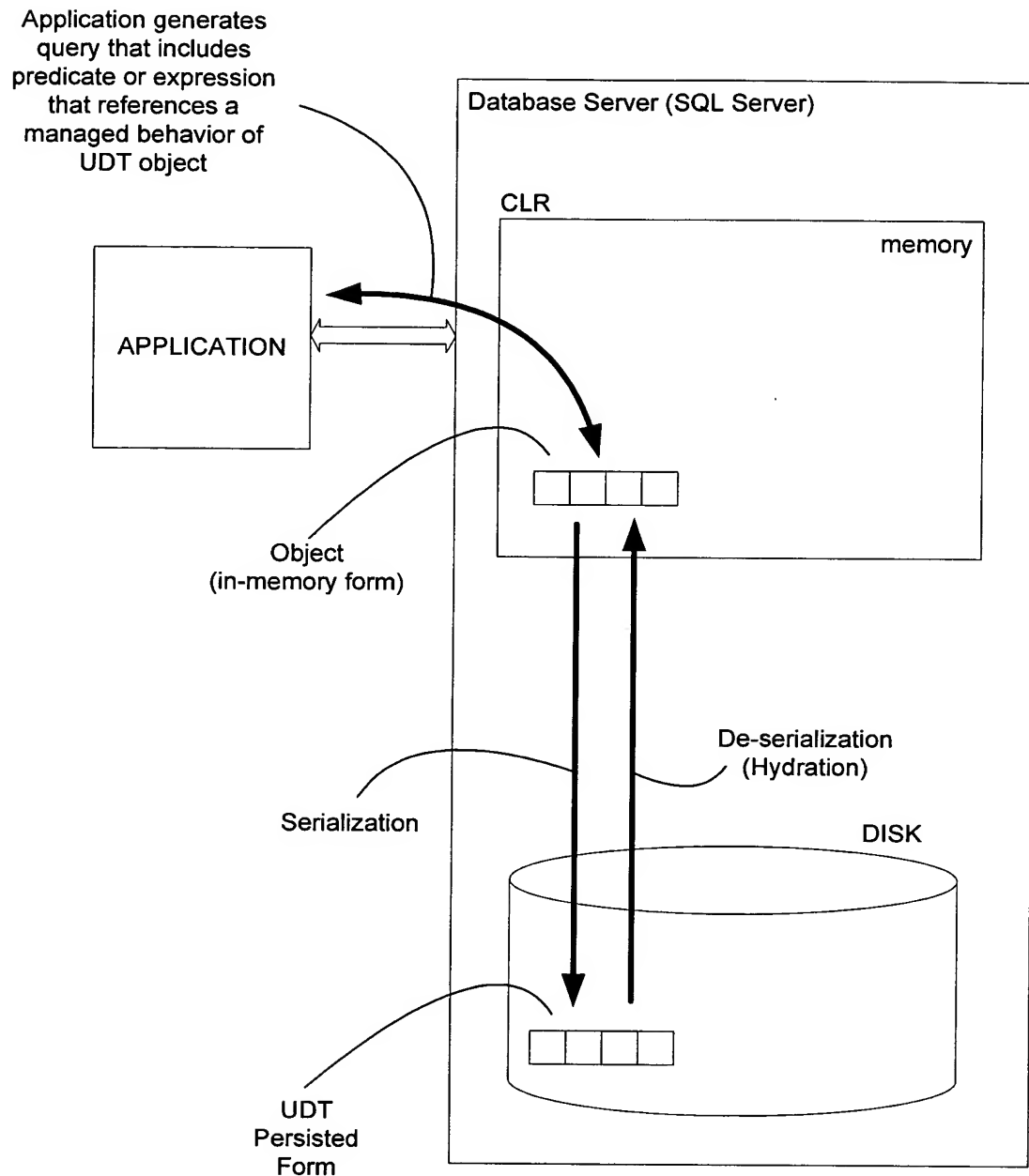


Fig. 2

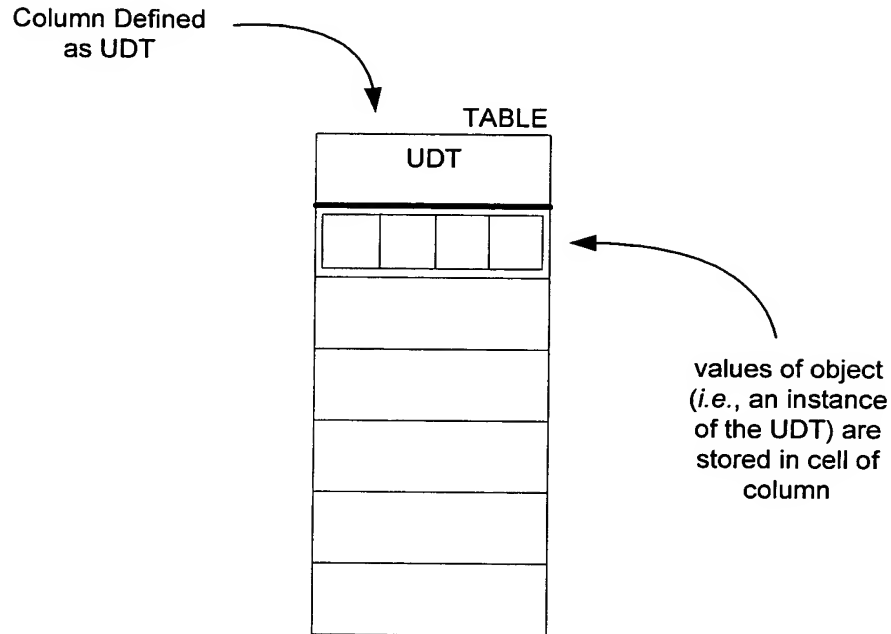


Fig. 3

<i>Member</i>	<i>Description</i>	<i>Return type</i>
SqlXml(stream)	Constructor	SqlXml
SqlXml(XmlReader)	Constructor	SqlXml
CreateReader()	Get (non-validating) XML reader	XmlReader

Fig. 4

```
Class Employee
{
    SqlString      fName;
    SqlString      lName;
    SqlDouble      Salary;
    SqlInt32       Age;
    SqlXml         Resume;
}
```

Fig. 5

```
Employee newEmp = new Employee();

// Assign values to the non-XML fields
newEmp.fName = "John";
newEmp.lName = "Doe";
newEmp.Salary = 50001;
newEmp.Age = 24;

// Obtain an XmlReader from a stream
StringReader stream = new StringReader("XML content here");
XmlTextReader reader = new XmlTextReader (stream);

// Assign value to the XML field Resume
newEmp.Resume = new SqlXml(reader);
```

Fig. 6

```
Employee Emp = <Employee instance>;
XmlReader reader1 = Emp.Resume.CreateReader();
XmlReader reader2 = Emp.Resume.CreateReader();
```

Fig. 7

```
Employee Emp = <Employee instance to be updated>;

// Update the non-XML fields
Emp.Salary = 51001;
Emp.Age = 25;

// Obtain an XmlReader from a stream
StringReader stream = new StringReader("new XML content
here");
XmlTextReader reader = new XmlTextReader (stream);

// Set the value of the XML field
Emp.Resume = new SqlXml(reader);
```

Fig. 8

```
Class TypedEmployee
{
    String    fName;
    String    lName;
    Float     Salary;
    Int       Age;

    [SqlUDTField (XmlSchemaCollection=
                  "myEmployeeSchema")]
    SqlXml    TypedResume;
}
```

Fig. 9

```
TypedEmployee newEmp = new TypedEmployee();

// Assign values for the non-XML fields
newEmp.fName = "John";
newEmp.lName = "Doe";
newEmp.Salary = 50001;
newEmp.Age = 24;

//*****
// Set up an XML schema set for validation
//*****
// Create an XML schema set
XmlSchemaSet mySchemaSet = new XmlSchemaSet();

// Add an XML schema with target namespace "Customer-ns" to
// mySchemaSet from "Customer.xsd" file
mySchemaSet.Add ("Customer-ns", "Customer.xsd");

//*****
// Write XML content into a validating writer
//*****
// Create stream to write into
TextStream stream = new TextStream();
StringWriter stmWriter = new StringWriter(stream);

// Create a validating writer
XmlFactory factory = new XmlFactory();
factory.WriterSettings.XsdValidate = true;
factory.WriterSettings.Schemas = mySchemaSet;

//*****
// Write into the validating writer
//*****
XmlWriter valWtr = factory.CreateWriter (stmWriter);
/* Write into valWtr */
valWtr.Close();

// Obtain an XmlReader from the stream
XmlReader reader = new XmlTextReader (stream);

// Set the value of the XML field
// Assign the XML content to the XML field
newEmp.TypedResume = new SqlXml(reader);
```

Fig. 10

```
// Get the type of class Employee
Type myType = typeof(TypedEmployee);

// Get the members of class Employee
MemberInfo[] myMembers = myType.GetMembers();

// Locate the "TypedResume" member and obtain the XML schema
// collection name
for(int i = 0; i < myMembers.Length; i++)
{
    If (myMembers[i].Name != "TypedResume")
        Continue;

    SqlUdtField mySqlAttribute =
        myMembers[i].GetCustomAttributes(SqlUDTField, true);
    string XmlSchemaColName =
        mySqlAttribute.XmlSchemaCollection;
}
```

Fig. 11

```
String XmlSchemaColName = "myEmployeeSchema" from class
definition (see Fig. 11)

// In-proc provider: database connection obtained from
// SqlConnection
// SqlConnection CurrentConnection =
//     SqlConnection.GetConnection();

// Out-of-proc provider: database connection must be provided
// by the application.
// SqlConnection CurrentConnection = SqlConnection in use

// Retrieve the XML schema collection from the server
SqlCommand cmd = CurrentConnection.CreateCommand();
cmd.CommandText =
    "SELECT NS.name, XML_SCHEMA_NAMESPACE(NS.name)
    FROM sys.xml_schema_collections SC
    JOIN sys.xml_namespaces NS
    ON (SC.xml_collection_id = NS.xml_collection_id)
    WHERE SC.name = '" + XmlSchemaColName;
SqlDataReader sqlRdr = cmd.ExecuteReader();

// Returned rowset contains two columns:
// target namespace URI (string), XML schema documents (SqlXml)

// Set up an XML schema set for validating reader
XmlSchemaSet mySchemaSet = new XmlSchemaSet();
While (sqlRdr.Read())
{
    mySchemaSet.Add (sqlRdr.GetString(0),
        // XmlReader from the SqlXml column
        sqlRdr.GetSqlXml(1).CreateReader());
}
```

Fig. 12

```
TypedEmployee Emp = <TypedEmployee instance>;

// Obtain (non-validating) XML reader
XmlReader nonValRdr =
TypedEmployee.TypedResume.CreateReader();

// Obtain the XML schema collection from the database
XmlSchemaSet mySchemaSet = XML schema coll. (See Fig.
12);

// Create a validating reader
XmlValidatingReader valRdr = new
XmlValidatingReader(nonValRdr);
valRdr.Schemas.Add(mySchemaSet);
valRdr.ValidationType = ValidationType.Schema;
valRdr.ValidationEventHandler += <some eventHandler>;

// Read ...
while (valRdr.Read()) {}
```

Fig. 13

```
TypedEmployee Emp = <TypedEmployee instance to be updated>;

// Update the non-XML fields
Emp.Salary = 51001;
Emp.Age = 25;

// Obtain the XML schema collection from the database
XmlSchemaSet mySchemaSet = XML schema coll. (See Fig. 12);

// Create a validating writer
TextStream stream = new TextStream();
StringWriter stmWriter = new StringWriter(stream);
XmlFactory factory = new XmlFactory();
factory.WriterSettings.XsdValidate = true;
factory.WriterSettings.Schemas = mySchemaSet;

// Write into the validating writer.
XmlWriter valWtr = factory.CreateWriter (stmWriter);
/* Write into valWtr */
valWtr.Close();

// Obtain an XML reader from the stream
XmlReader reader = new XmlTextReader (stream);

// Assign the XML content to the XML field
TypedEmp.TypedResume = new SqlXml(reader);
```

Fig. 14


```

class Employee
{
    qlString      fName;
    SqlString     lName;
    SqlDouble     Salary;
    SqlInt32      Age;
    SqlXml        Resume;

    // Apply XSL transformation to the Resume field according
    // to a specified XSL file (XslFilePath)

    public static SqlXml TransformXml(SqlString XslFilePath)
    {
        XmlDocument xslDoc = new XmlDocument();
        xslDoc.Load (XslFilePath);
        XPathNavigator xslNav = xslDoc.CreateNavigator();

        // XSL transformation to apply
        XslTransform xform = new XslTransform();
        xform.Load (xslNav,null);

        // Obtain navigator for the Resume (XML) data
        XPathDocument xPathDoc =
            new XPathDocument (Resume.CreateReader());

        XPathNavigator nav = xPathDoc.CreateNavigator();

        // Apply transformation
        XmlReader    xFormReader = xform.Transform(nav, null);

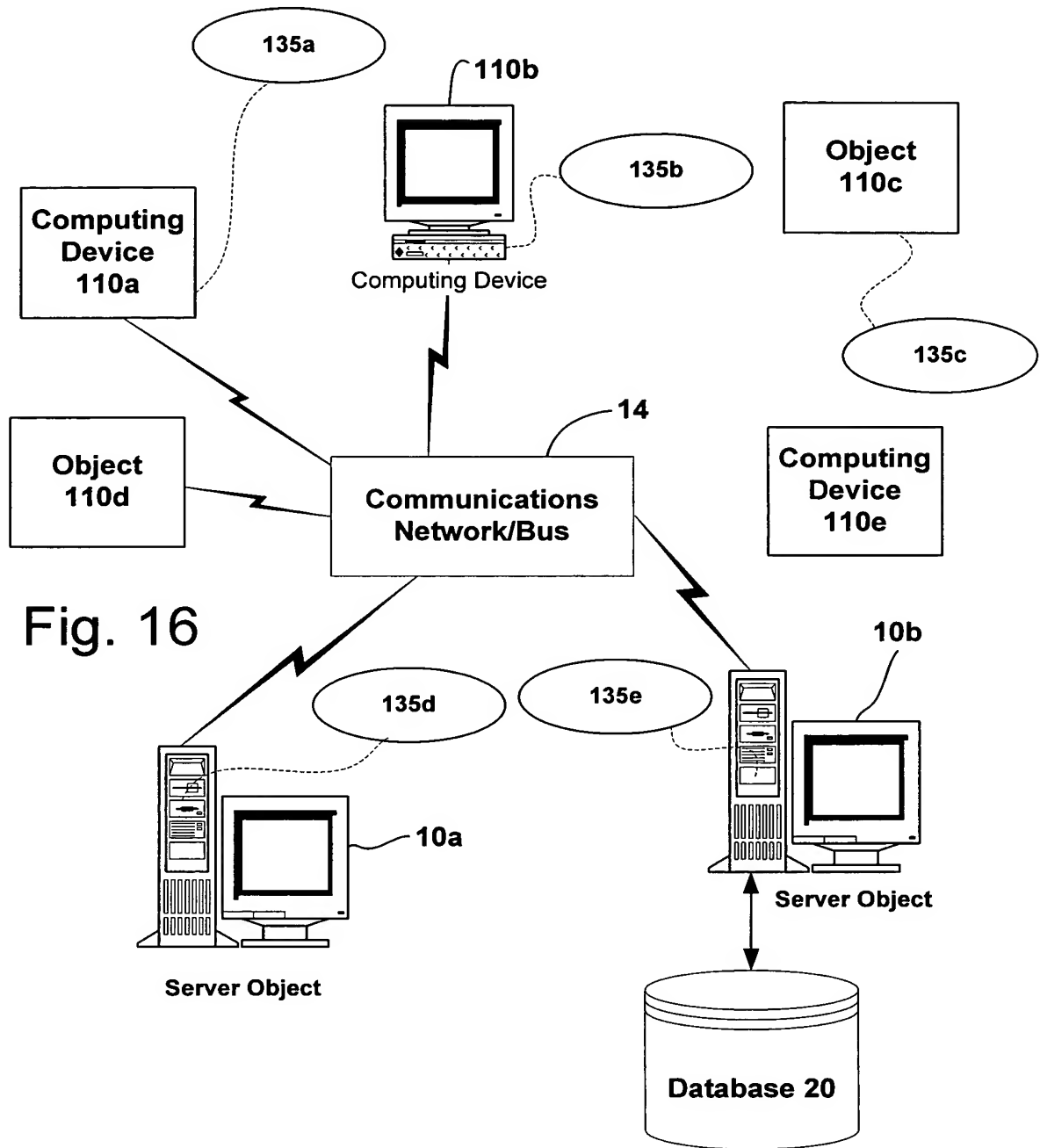
        // Return the transformed value
        return new (SqlXml(xFormReader));
    }
}

=====

SELECT      udtEmp.TransformXml ('C:\Employee\Transform.xsl')
FROM        tabEmployee

```

Fig. 15



Computing Environment 100

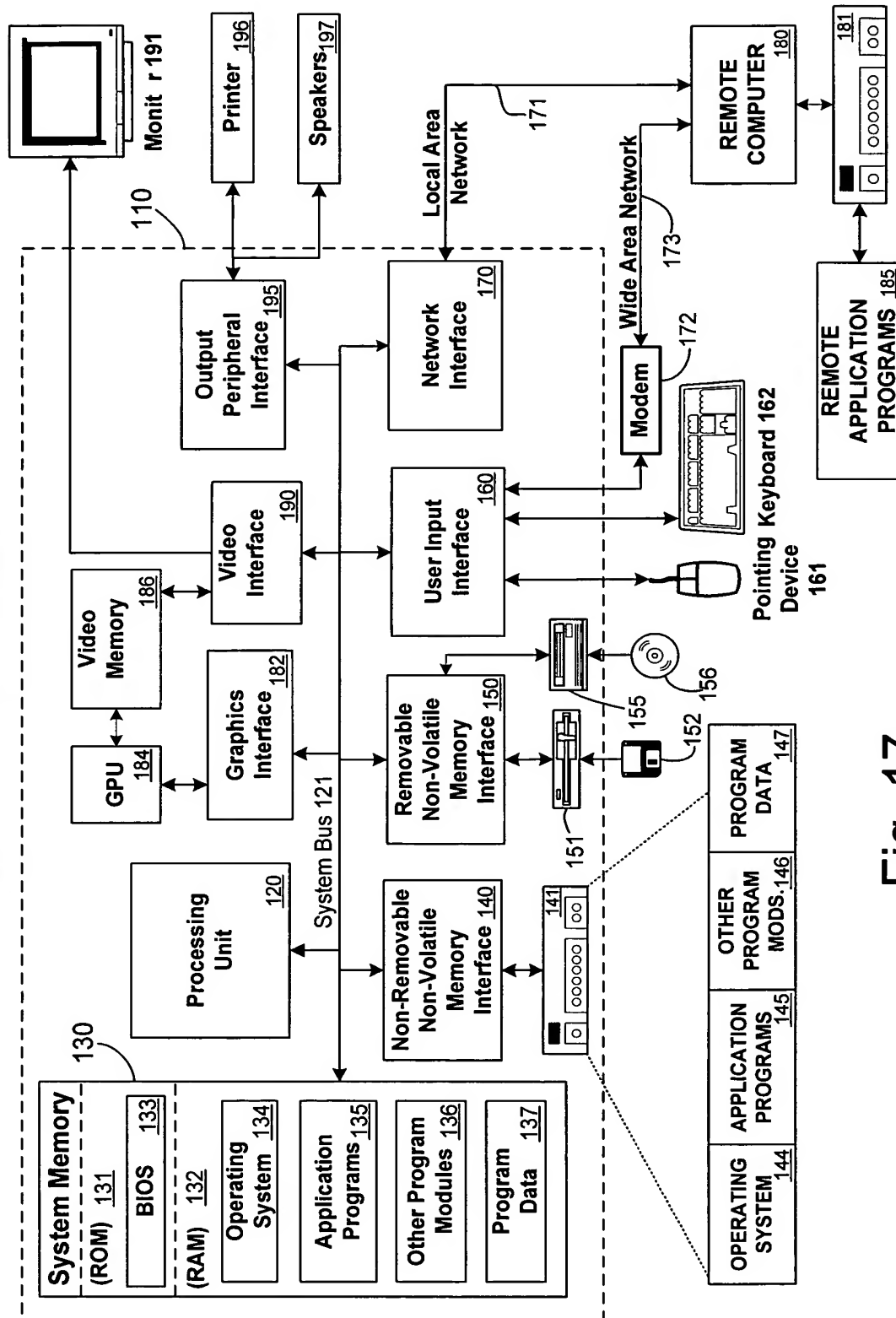


Fig. 17